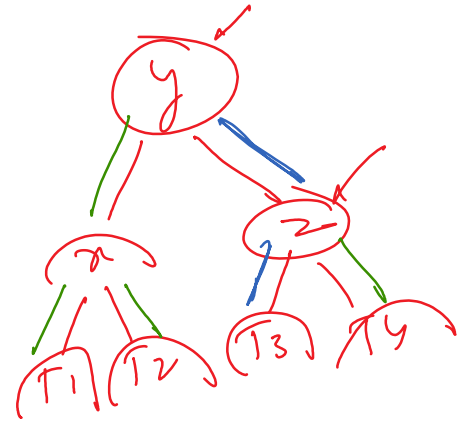
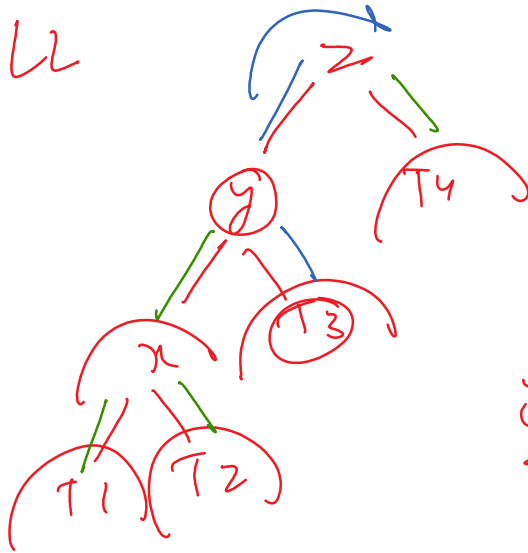


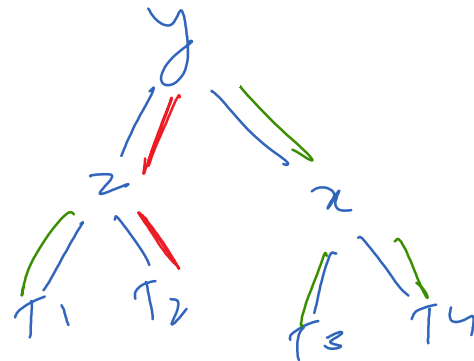
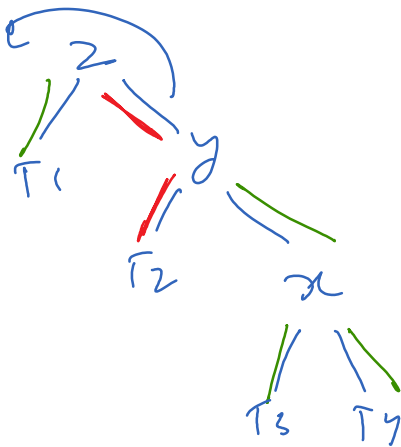
LL



$y = z.l$
 $t3 = y.r$
 $z.l = t3$
 $y.r = z$

x y

RR

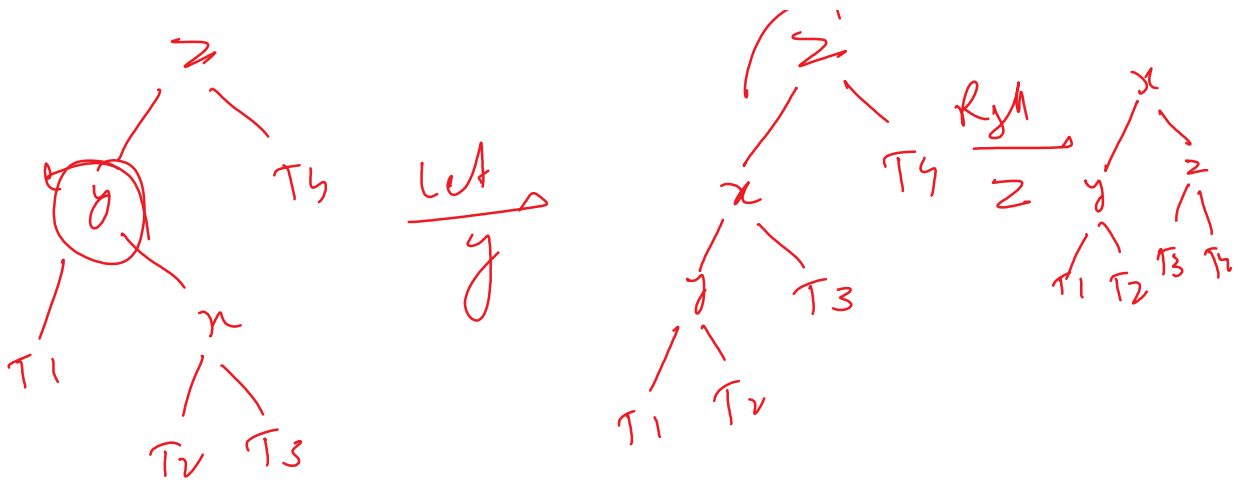


LR

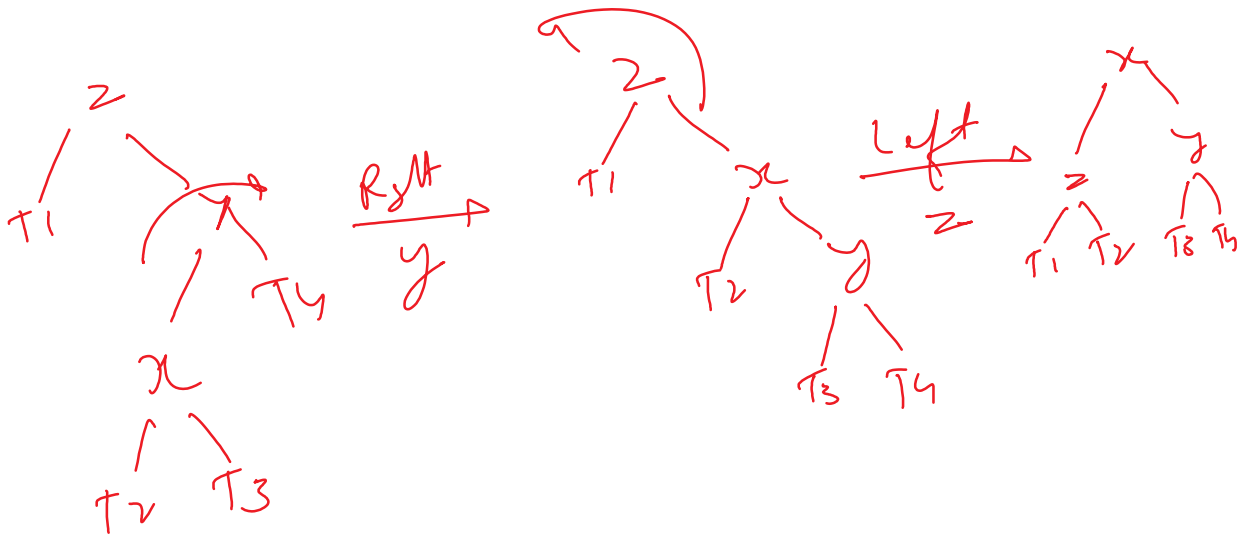


o.k

x



RL

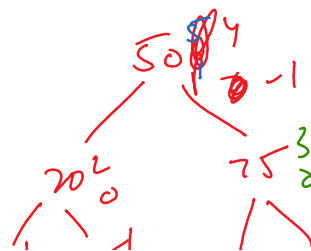


11:10 - 11:20

add, lr, rr

20 <- 50[4, -1] -> 75 ✓
 12 <- 20[2, 0] -> 25 ✓
 . <- 12[1, 0] -> . ✓
 . <- 25[1, 0] -> . ✓
 62 <- 75[3, 0] -> 87 ✓
 60 <- 62[2, 0] -> 70 ✓
 . <- 60[1, 0] -> . ✓
 . <- 70[1, 0] -> . ✓
 80 <- 87[2, 0] -> 90 ✓
 . <- 80[1, 0] -> . ✓

LL
 LR
 RL



rr
 n.b < -1

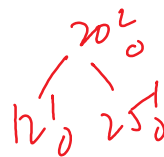
```

.<- 60[1, 0] -> .
.<- 70[1, 0] -> .
80 <- 87[2, 0] -> 90
.<- 80[1, 0] -> .
.<- 90[1, 0] -> .

```

From
<https://pepcoding.com/resources/online-java-foundation/binary-tree/size-sum-max-height-binary-tree-official/ojquestion#>

l
 $n.b > 1$
 $n.l.b \geq 0$



rr
 $n.b < -1$
 $n.r.b < 0$

lr
 $n.b > 1$
 $n.l.b < 0$

rr
 $n.b < -1$
 $n.r.b \geq 0$

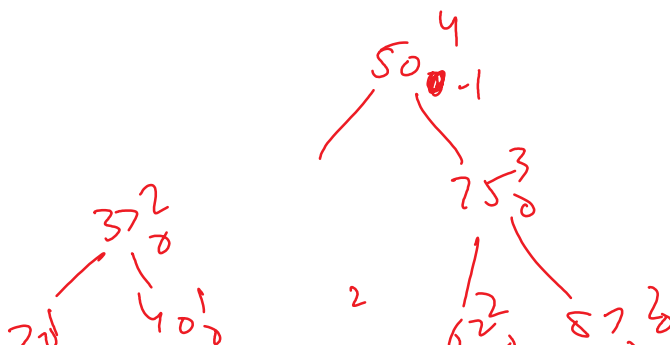
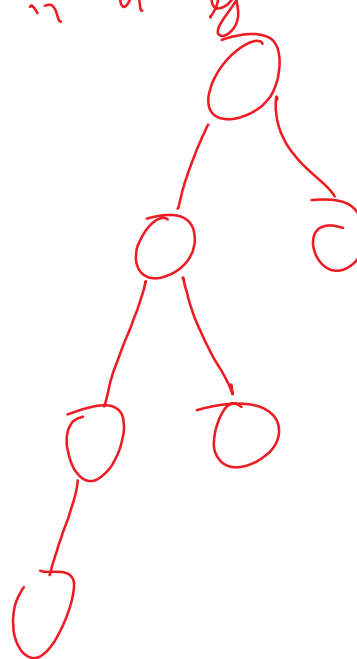


$bal > 0$ left side heavy
 $bal < 0$ right side heavy

```

// bal = lh - rh
if (node.bal > 1) { // ll, lr
    if (node.left.bal >= 0) { // ll
        node = rightRotate(node);
    } else { // lr
        node.left = leftRotate(node.left);
        node = rightRotate(node);
    }
} else if (node.bal < -1) { // rr, rl
    if (node.right.bal < 0) { // rr
        node = leftRotate(node);
    } else { // rl
        node.right = rightRotate(node.right);
        node = leftRotate(node);
    }
}

```



30_0 40_0

2

62_0 87_0
 60_0 70_0 80_0 90_0